SYSTEM AND METHOD FOR PROVIDING INTELLIGENT PRE-STAGING OF DATA IN A COMPUTE ENVIRONMENT

PRIORITY CLAIM

[0001] The present application claims priority to U.S. Provisional Application No. 60/552,653 filed March 13, 2004, the contents of which are incorporated herein by reference.

RELATED APPLICATIONS

[0002] The present application is related to Attorney Docket Numbers 010, 0011, 010-0011A, 010-0011B, 010-0011C, 010-0013, 010-0026, 010-0028 and 010-0030 filed on the same day as the present application. The content of each of these cases is incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0003] The present invention relates to managing job submissions in a compute environment such as a cluster and more specifically to intelligent data just in time data pre-staging to optimize the use of diverse compute resources.

2. Introduction

[0004] The present invention relates to a system and method of allocation resources in the context of a grid or cluster of computers. Grid computing may be defined as coordinated resource sharing and problem solving in dynamic, multi-institutional collaborations. Many computing projects require much more computational power and resources than a single computer may provide. Networked computers with peripheral resources such as printers, scanners, I/O devices, storage disks, scientific devices and instruments, etc. may need to be coordinated and utilized to complete a task.

[0005] Grid/cluster resource management generally describes the process of identifying requirements, matching resources to applications, allocating those resources, and scheduling and monitoring grid resources over time in order to run grid applications as efficiently as possible. Each project will utilize a different set of resources and thus is typically unique. In addition to the challenge of allocating resources for a particular job, grid administrators also have difficulty obtaining a clear understanding of the resources available, the current status of the grid and available resources, and real-time competing needs of various users. One aspect of this process is the ability to reserve resources for a job. A cluster manager will seek to reserve a set of resources to enable the cluster to process a job at a promised quality of service.

[0006] General background information on clusters and grids may be found in several publications. See, e.g., Grid Resource Management, State of the Art and Future Trends, Jarek Nabrzyski, Jennifer M. Schopf, and Jan Weglarz, Kluwer Academic Publishers, 2004; and Beowulf Cluster Computing with Linux, edited by William Gropp, Ewing Lusk, and Thomas Sterling, Massachusetts Institute of Technology, 2003.

[0007] It is generally understood herein that the terms grid and cluster are interchangeable in that there is no specific definition of either. In general, a grid will comprise a plurality of clusters as will be shown in FIG. 1. Several general challenges exist when attempting to maximize resources in a grid. First, there are typically multiple layers of grid and cluster schedulers. A grid 100 generally comprises a group of clusters or a group of networked computers. The definition of a grid is very flexible and may mean a number of different configurations of computers. The introduction here is meant to be general given the variety of configurations that are possible. A grid scheduler 102 communicates with a plurality of cluster schedulers 104A, 104B and 104C. Each of these cluster schedulers communicates with a respective resource manager 106A, 106B or 106C. Each resource manager communicates with a respective series of compute resources shown as nodes 108A, 108B, 108C in cluster 110, nodes 108D, 108E, 108F in cluster 112 and nodes 108G, 108H, 108I in cluster 114.

[0008] Local schedulers (which may refer to either the cluster schedulers 104 or the resource managers 106) are closer to the specific resources 108 and may not allow grid schedulers 102 direct access to the resources. Examples of compute resources include data storage devices such as hard drives and computer processors. The grid level scheduler 102 typically does not own or control the actual resources. Therefore, jobs are submitted from the high level grid-scheduler 102 to a local set of resources with no more permissions that then user would have. This reduces efficiencies and can render the reservation process more difficult.

[0009] The heterogeneous nature of the shared resources also causes a reduction in efficiency. Without dedicated access to a resource, the grid level scheduler 102 is challenged with the high degree of variance and unpredictability in the capacity of the resources available for use. Most resources are shared among users and projects and each project varies from the other. The performance goals for projects differ. Grid resources are used to improve performance of an application but the resource owners and users have different performance goals: from optimizing the performance for a single application to getting the best system throughput or minimizing response time. Local policies may also play a role in performance.

[0010] An administrator can partition a cluster and identify a set of resources to be dedicated to a particular purpose and another set of resources can be dedicated to another purpose. In this regard, the resources are reserved in advance to process the job. To illustrate, an example is provided.

Assume that the weather bureau needs to do a compute intensive hurricane analysis. They will desire to gather a large amount of stored data from disk and then process that data and store the resulting computed data. A scheduler, to manage the cluster resources for this job, will schedule the disks to retrieve the data, network routers with an appropriate bandwidth to transmit the data, computer processors to then process the data, and then network routers and data disks to transmit and store the computed data. The availability of the disks for these retrieval and storage aspects of the job may not overlap specifically in time with the time for the availability of the computer processing or transmission resources.

[0011] To manage the jobs submissions, a cluster scheduler will employ reservations to insure that jobs will have the resources necessary for processing. Figure 1B illustrates a cluster/node diagram for a cluster 124 with nodes 120. Time is along the X axis. Node 1 has a reservation on it and an access control list (ACL) 122 which is static. The ACL 122 is based on the credential available to the requestor or person submitting the job. In other words, the user, group, the account, the class or quality of service the requestor has and/or is asking for. The job either will get onto the ACL 122 based on the criteria or it won't. That determination is made at the time the job is submitted for entry on the ACL 122.

[0012] The approach described above for reserving and processing jobs utilizing the various cluster resources has drawbacks in efficiency. The retrieved data from the disk storage resource may not coincide with the computer processing resources. In other words, the data may be retrieved from disk but the computer processors may not be ready to process the data given the other jobs submissions that are operating within their reservations on the cluster resources. To improve the management of cluster resources, what is needed in the art is an improved method for managing the consumption of diverse resources within a compute environment such as a cluster or grid.

SUMMARY OF THE INVENTION

[0013] Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth herein.

[0014] A system and method for performing intelligent data pre-staging for a job submitted to a cluster environment. The method aspect comprises determining availability of compute resources including availability timeframes to process the submitted job, determining data requirements for

processing the job and determining a co-allocation in time reservation.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0016] FIG. 1A illustrates generally a grid scheduler, cluster scheduler, and resource managers interacting with compute nodes;

[0017] FIG. 1B illustrates an access control list controlling access to a group of nodes within a cluster environment;

[0018] FIG. 2 illustrates the interaction between a compute reservation and a data reservation;

[0019] FIG. 3 illustrates the data stage-in process and data stage-out process with a compute reservation; and

[0020] FIG. 4 illustrates a method aspect of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0021] Various embodiments of the invention are discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without parting from the spirit and scope of the invention.

[0022] The present invention applies to a compute environment examples of which include clusters and grids. It is preferable that is be used to manage cluster resources but there is no requirement that it be limited to that context. The cluster may be part of a data center, host facility, virtual hosting center, utility-based computing environment and so forth. The present invention applies to any scenario where there is a need for compute resource guarantees with time offsets. In other words, a hosting center may have a service level agreement with a company to provide a certain amount of compute resources within two hours of a request for resources.

[0023] The particular scenario where the invention applies is where a job submission requires a staging of data, which typically involves retrieving the data from disk and storing the data in a cache in preparation for compute resources to become available to process the data. In the cluster

environment, the compute resources will be processing other jobs and the data staging is to enable the compute resources to finish other prior commitments and be ready to process the job associated with the staged data.

[0024] The invention comprises a system, method and computer-readable media for performing a data pre-staging to analyze the resources and the data to reduce any wasted resources when diverse resources such as storage disks, cache, compute resources, and transmission bandwidth must all be reserved and used to complete a job. The "system" embodiment of the invention may comprise a computing device that includes the necessary hardware and software components to enable a workload manager or a software module performing the steps of the invention. Such a computing device may include such known hardware elements as one or more central processors, random access memory (RAM), read-only memory (ROM), storage devices such as hard disks, communication means such as a modem or a card to enable networking with other computing devices, a bus that provides data transmission between various hardware components, a keyboard, a display, an operating system and so forth. There is no restriction that the particular system embodiment of the invention has any specific hardware components and any known or future developed hardware configurations are contemplated as within the scope of the invention when the computing device operates as is claimed.

[0025] FIG. 2 illustrates an ideal interaction between compute resources 202 and data resources 204. Time is along the x axis in this figure. This interaction is mentioned as being ideal because what is illustrated in FIG. 2 is the scenario where the compute reservation of resources (node processors or other compute resources) and the data resources (such as storage disks) are both concurrent in time. In this case, since the reservations of resources span the same time frame, the resources are always available for each other for job processing. In other words, if the compute nodes need data at any time during the processing of a job, the data resources will always be reserved and available for providing the data stage-in or data stage-out necessary. This is not always the most efficient use of resources, however.

[0026] FIG. 3 illustrates an aspect of the invention wherein the data stage-in reservation is made 302 earlier in time to the compute reservation 304. Other compute reservations exist 308 before and after the current reservation. The data stage-in reservation of data resources is timed to overlap the compute reservation an appropriate amount of time to provide the necessary data to the compute resources for processing the job. Then the data resources are reserved for another entity 310 since these resources will not be needed until the data stage-out reservation 306 which may involve, for example, receiving the processed data from a weather analysis of a hurricane. FIG. 3 illustrates a more advanced and efficient use of resources.

[0027] With the principles in mind above, the steps of the invention will be explained next with reference to FIG. 4. The method may be performed by a cluster scheduler, or grid scheduler, or other software component associated with the management of resources in the cluster environment. Therefore, any of these components may be considered the "system" that performs the steps of the method embodiment of the invention.

[0028] A reservation of resources is made or a job is submitted for processing on the cluster. In order to actually do intelligent data pre-stage and co-allocation of resources and time, the first step in intelligent data pre-stage is the analysis of time to stage data. The system must determine how long it's going to take the complete the particular task by estimating that timeframe based on network information, network speed, faults, statistical fluctuation, delivered bandwidth by the network, size, and any issues, you basically have to ramp up the initialize step, a data transfer step, and a prologue step, a termination step which completes the record and verifies the successful transfer of data. In this regard, the method comprises identifying compute resources to process the job and locating various timeframes in which those resources have availability (402). This is the first step related to the compute resources. The system evaluates the data requirements and resources that the job would consume in terms of quantity of data and in terms of speed of migration of that data (404). This is the second step related to the data and network resources. Once the rate of data transfer is identified, the system determines the timeframe by which the data staging would need to make it available (406). The goal is to maximize the timing of the allocation of resources between the network bandwidth, the data cache or disk usage, and the compute resources. The allocation of the data cache and network bandwidth occurs earlier in time followed by the compute resources. There also is likely some data caching or bandwidth needs for post-processing transmission and storage of data.

[0029] For an input file, one could optimize resources by releasing the data resources after some time into the job once the job has successfully loaded all that information into memory. Whether or not that's actually done would depend on how highly constrained the data resources were. Basically, there would be a requirement to start a data stage and some time offset from when the compute cycle begins. Sometime after that compute cycle is over, the system allocates another data stage for stage back or transmission of the processed data.

[0030] The present invention improves the efficiency of the data in-gathering stage where those resources are not wasted by the mis-timing of the gathering and processing resources. The invention involves timing the gathering of data with the availability of compute resources to process the data. Typically, the compute resources are most constrained by reservations, which enable an administrator to over constrain data and network resources without an overall impact on utilization.

[0031] Next, the system performs a series of calculations to evaluate existing resource guarantees and reservations already in place to create a range list. A range list indicates all the availability time frames. With all the available time frames, the system calculates, based on incorporated duration information and the availability time frames information, which available time frames the request could actually start. For example, if one had resources available for a period of two hours and had a request that lasted one hour, the time during which that request could start is only during the first hour of that availability.

[0032] The system converts the availability range to a start range and once that completes, the system then performs the same evaluation for the second request in which the system performs the same process independently to evaluate when resources are available and converts that availability information into start information. This process may occur for any number of n requests. The various requests may relate to different types of resources. For example, one request may be processed for compute resources and another request for data resources, provisioning resources or bandwidth resources etc.

[0033] Once all the requests have been converted to start ranges, the system shifts the start ranges by the offset and performs an intersection operation (an AND operation) on the combination start range. With the intersection, the system shifts it back by the negative of the offset the resulting information provides when to start each reservation. Like any intersection operation, there will probably be multiple viable solutions that the system presents to the external system making the requests. The system could present the invention as a number of start time availabilities. Once a start time is selected by an administrator or use, the system shifts everything back and reserves the resources during those time frames.

[0034] Once you have that time estimate done and performed steps as set forth above, the method comprises creating a co-allocation in time reservation (408). The key to this process is determining a number of calculations based on: (1) the duration and quantity of the first compute resources, (2) the duration and quantity of the second data and network bandwidth resources, (3) the fact that the second step must complete prior to the beginning of first step, (4) the job execute within certain constraints, (5) the offset time. With this information, the system performs a co-allocation reservation in which the system requests the resources for whatever the first step in time is. So in this case, the system determines the information for the data migration.

[0035] Within the workload manager of the present invention, the system can actually pass back transaction IDs associated with a co-allocation in time reservation. The transaction ID can then be used as a reference to the particular analysis or resulting reservation. So when a user submits a query they can have a concept of a transaction ID associated with that query. The transaction ID indicates that a person has this particular query subject to certain constraints and they know there is

a certain block of resources available. They can mask the specifics of the query and if they want to come back and get these resources they simply indicate that they would like to commit the particular transaction under the covers, once all the resources are done.

[0036] Embodiments within the scope of the present invention may also include computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code means in the form of computer-executable instructions or data structures. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or combination thereof) to a computer, the computer properly views the connection as a computer-readable medium. Thus, any such connection is properly termed a computer-readable medium. Combinations of the above should also be included within the scope of the computer-readable media.

[0037] Computer-executable instructions include, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. Computer-executable instructions also include program modules that are executed by computers in stand-alone or network environments. Generally, program modules include routines, programs, objects, components, and data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represents examples of corresponding acts for implementing the functions described in such steps.

[0038] Those of skill in the art will appreciate that other embodiments of the invention may be practiced in network computing environments with many types of computer system configurations, including personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. Embodiments may also be practiced in distributed computing environments where tasks are performed by local and remote processing devices that are linked (either by hardwired links, wireless links, or by a combination thereof) through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0039] Although the above description may contain specific details, they should not be construed as limiting the claims in any way. Other configurations of the described embodiments of the

invention are part of the scope of this invention. Accordingly, the appended claims and their legal equivalents should only define the invention, rather than any specific examples given.